POCKET COMPUTER NEWSLETTER

© Copyright 1982 — Issue 18 October

NEW CASIO FX-700P CRACKS $100.00 PC PRICE BARRIER

Just one year after Casio introduced the first "under $200.00" PC, it has scored another price breakthrough. The new PC, designated the FX-700P is slated for U.S. deliveries in October. It is priced at $99.95.

The unit appears to be a slightly smaller-sized version of the 702P, with some features of the earlier unit dropped.

One highly noticeable change is in the keyboard layout. It is now in the familiar QWERTY arrangement, with a numeric keypad along the right side. Another difference is a reduction in the size of the LCD, down to just 12 characters plus a battery of annunciators.

The unit measures 3/8 inches thick by 6-1/2 inches wide and 2-3/4 inches deep. It may also set a new weight record coming in at a mere 4.2 ounces when equipped with two CR2032 lithium batteries. Battery life is said to be about 300 operational hours.

User memory available in the unit is slightly less than the 702P. The FX-700P sports 1,568 program steps or 222 memories. Memory may be partitioned in steps to allocate variables storage space and program storage. Up to 10 programs may be identified and accessed within those memory boundaries. There is also room provided for a 30-character string element.

The BASIC language operating system includes the statements and commands: INPUT, PRINT, GOTO, FOR-NEXT, IF-THEN, GOSUB, RETURN, STOP, END, RUN, LIST, LIST A, MODE, SET, VAC, CLEAR, CLEAR A, DEFM, SAVE, SAVE A, LOAD, LOAD A, PUT, GET and VER. Program functions include: KEY, CSR, LEN, MID and VAL. Program stacks permit up to 8 levels of subroutines, 4 levels of FOR-NEXT loops, 6 levels of numerical values and 12 levels of calculation elements.

The unit also sports a number of mathematical functions such as: choice of degree/radian/gradient angular units, log and exponentials, square root, powers, integer, absolute value, pi, random numbers and others.

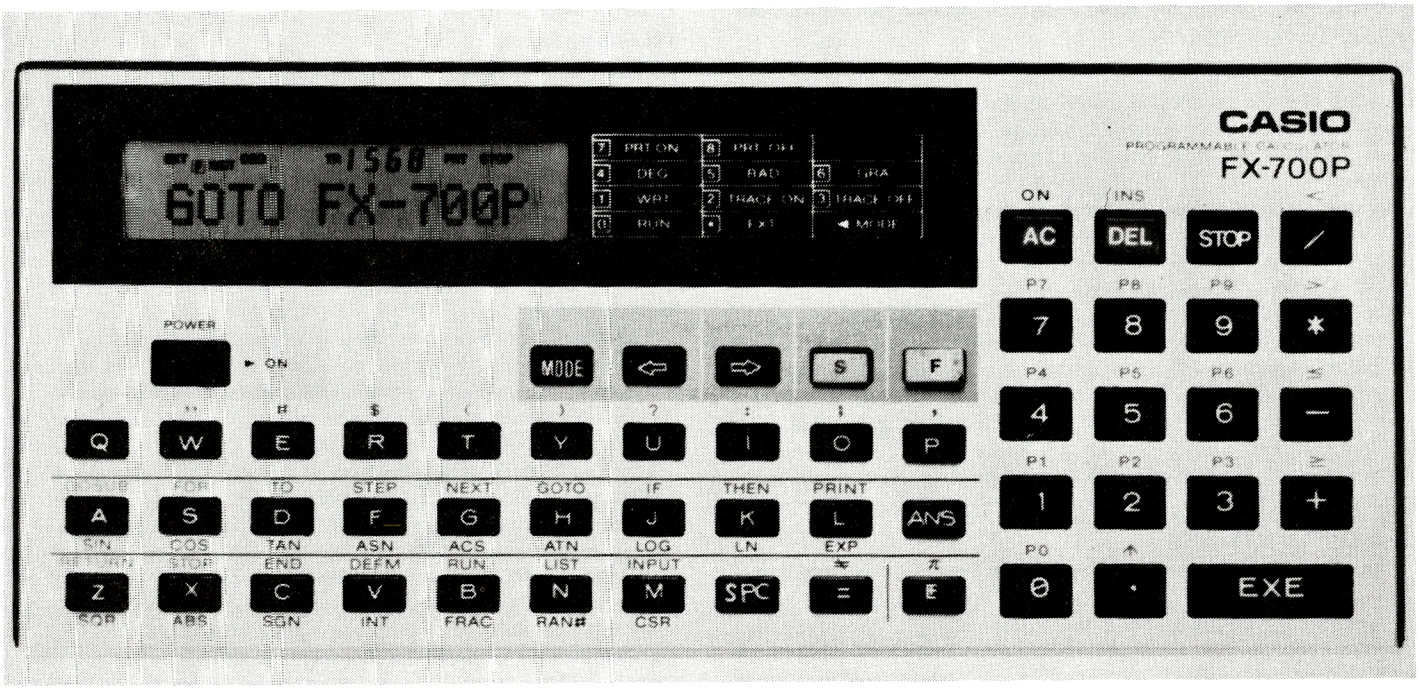
The liquid-crystal display with 12 character positions can display the mantissa of a number to 10 positions (including sign) or display an 8-digit mantissa plus a 2-digit exponent. The display also is capable of displaying both upper- and lower-case alphabetical characters plus a number of symbols. Special annunciators along the top of the display provide additional information about the status of the PC.

The keyboard achieves versatility be providing as many as three functions to a key. Special shift keys determine which function a key will serve when it is pressed. BASIC statement keywords have been assigned to keys to permit rapid entry of programs. The 54-key unit reportedly uses the standard ASCII character coding convention.

An optional tape recorder interface, called the FA-3 is reported as being available soon, at low cost.

A program library offering a selection of typical application programs is supplied with the unit. For more information contact: *Casio, Consumer Products Division, 15 Gardner Road, Fairfield, NJ 07006.*

Photo *Casio FX-700P Pocket Computer Programs in BASIC.*



THE HP-75: A FORMIDABLE COMPANION

I hardly know where to start telling you about Hewlett-Packard's entry into the true portable computer market. Let me simply preface my review by saying that this machine will very likely set a standard. One against which other makers of PCs will find themselves compared.

It is obvious that this machine was designed by a team that did their homework. They have managed to combine practically every desirable feature that experience has taught pocket computer users they need, into the HP-75. What is astounding is that they managed to do it on their first try!

Size

Hewlett-Packard does not call the HP-75 a *pocket* computer, they bill it as a highly *portable* computer. They are right. At 5 by 10 inches and 1-1/4 inch thick it is just a little too big to go in a pocket. But, at just a pound and a half, it feels and carries like a slim-line book. When snuggled safely in its leather carrying case (supplied as standard with the unit) you might think someone was toting an electronic multimeter, instead of a fully-functional computer! Oh yes, there is also room in the case to store a good supply of the little magnetic strip-cards which can hold programs or data. The card reader is built right in to the HP-75. This arrangement sure beats trying to stuff a pocket computer, a portable tape recorder, half a dozen cassettes and a bunch of connecting cords into a briefcase just so one can quickly exchange programs or data. In other words, the HP-75 adds a significant measure of *practicality* to the concept of portable computing – despite the fact that the unit doesn't actually fit into a pocket!

The size of the unit is just large enough so that the keyboard can be comfortably arranged for touch-typing. There are a few features of this keyboard that I personally consider drawbacks. One is that the keys are shaped like and feel like "calculator" buttons. I hope that in future models the designers can make the "feel" of this keyboard just a little better for touch-typists.

A second feature that I find a little disconcerting is the manner in which the shift key operates. You must hold it depressed while you press the key that is to be shifted. While this is exactly as one uses a typewriter, I guess I have been spoiled by the action of the Sharp/Radio Shack PCs. In those units, one press of the shift key means the next regular key pressed will be shifted – so you don't have to hold the shift key down. You hunt-and-peckers will have to get used to using both hands on a HP-75!

Taken in over-all context, however, these negative aspects of the keyboard's operation are minor. The fact is, you *can* touch-type on this keyboard with considerable speed – something you really have not been able to do on other PCs to date.

Features – Features – Features

There are so many features on the HP-75 that it will not be possible to cover them all (by a long shot) in this review. But, I will try to give you a feel for the type of capabilities that have been provided on this machine.

To start with, let me point out that the unit comes standardly equipped with 48 kilobytes worth of operating system in ROM plus 16K of user RAM. Now, 48K of ROM buys you a lot of operating system – and HP capitalized on every bit of it.

(One quick aside here. That 48K operating system in ROM *does* eat up some of the user's RAM – about 2K worth, typically. This is used for pointers, counters, I/O buffers, etc. This leaves about 14K for user programs in a standard system. You can add an optional 8K RAM module to form a 24K system, of which about 22K is available for user programs and data. I mention this because an HP press release I saw could leave the impression that the operating system does not require any user RAM.)

The Appointment Mode

To begin with, the system incorporates an extremely sophisticated "appointment reminder" capability. How sophisticated? Well, you can schedule appointments under the current one year period. Or, if that is not enough, you can use the extended 10,000 year (you should be so lucky!) calendar.

Pressing an 'APPT' key brings up a template on the LCD (liquid-crystal display) screen. You key in the date and time of your appointment using this template. If you don't know certain parameters (such as the day of the week on which a certain date falls), the operating system figures that information out for you. You also tell the system what type of audible alarm you want provided (there are ten possibilities) and whether the appointment is singular or repetitive. (If repetitive, you specify how often the appoinment repeats.) You can also enter a message that describes the nature of the appointment as a reminder to yourself.

Once appointments have been keyed in, they are automatically announced *regardless of the state of the machine!* Yes, even if the machine is shut off. It will turn itself on, issue an audible alert (if you requested an audible option), turn on a visual annunciator on the LCD screen, and even proceed to execute a program, if that was desired!

The system operates such that it holds and alerts you to any appointments that are past-due every time you use the unit. It will do this until you acknowledge or otherwise remove the appointment from the system's memory!

Depending on which type of alarm you specify, an appointment reminder can range from the subtle (just a little annunciator displayed on the screen) to the not-so-subtle (how about a wailing siren every 15 seconds?)!

There are lots of features of this appointment system that I don't have the space to go into detail on. Let me say that I have spent several hours "playing around" with the various capabilities of just this mode of the unit and I still don't know all of what it is capable of doing.

Oh yes, you say you might have large programs to run and you do not want to waste any of your valuable memory with appointment schedules? No problem. All the appointments are held in a "file." That file, like any other file in the system, can be offloaded to a mass storage device and reloaded any time! I will have more to say about files....

The Time, Please!

How does the HP-75 keep track of the arrival of appointments? Why, with its real-time clock, of course. And what a clock. Do you know of any other PC that allows you to adjust the accuracy of its real-time clock?

This clock can be personalized to suit your own preferences. You can set it for 12- or 24-hour format. You can even arrange the Day/Month/Year template to your own liking. The clock can be used to activate the unit or external devices, called by programs to record actual times at which events occur, etc. And, as I said above, you can calibrate and increase or decrease the speed of the clock.

The Editing System

Up until the HP-75C, the Sharp/Radio Shack PCs had, in my opinion, the best editing capabilities of the portables. Virtually every other manufacturer of PCs had slipped in this area by leaving out vital editing functions. Such as, not providing the ability to scroll backwards (from a high numbered line to a lower numbered line). The HP-75C has virtually every feature of the, say, Sharp PC-1500, plus many more.

To start with, editing begins at the *file* level. Hewlett-Packard defines a "file" as "an area of memory that can be identified by name and manipulated as a unit." Oh yes, you can have many separate files stored in memory at any time.

To keep track of your files, there is a cataloging function built into the operating system. Any time you want to review what files you have in memory, you just ask for the system catalog. It will summarize all files by their names, type, amount of memory each consumes and the time and date at which each was established.

Files can be duplicated in memory or transferred (unless secured) to external devices (including magnetic strip-cards).

There are several different types of files. Two of the most commonly used types are termed BASIC and TEXT files. Yes, one is intended primarily for the development of programs, the other for the storage of plain text files. But, would you believe, there is a TRANSFORM command that enables you to convert one type of file into the other!

Once you have established a file, there are all kinds of editing functions at various levels. For instance, an automatic line numbering

option is available. Even more valuable to program developers is the BASIC renumbering capability. You can specify the renumbering of all or just part of a BASIC program. A very nice capability to have provided on a PC.

The sophistication continues down to the character level. A set of editing keys provide complete line and within-line scrolling capability. That is, you can scroll backwards and forwards through complete lines in a listing. Or, you can scroll across a single line. Shift options enable you to instantly jump from the beginning to the end of a line and vice versa. Similarly, a "fetch" key enables you to quickly specify a particular line within a listing or instantly return to the last line you were working on. Other editing keys provide the ability to insert or delete individual characters within a line.

In summary, the general file, text and program editing capabilities of this machine are superb! If you have any problem in this area, it will simply be in *learning* about all of the editing capabilities that are available.

My question, after examining the HP-75C, is why haven't the desktop computers had these kinds of capabilities provided as standard years ago? How many *desktop* units that you know of will let you *merge* files in memory? Yes, you can even do that (whole merges or partial-file merges) on the HP-75C!

It's a Calculator, too!

One hardly has to point out that any HP product would automatically include a calculator, but I will mention some of the types of functions that are provided as standard on this product.

You can assign values to variables directly from the keyboard. You can specify real, short form or integer variables. You can extract absolute, integer part, fractional part or integer values. You can set "floor" or "ceiling" values in calculations. You can find the square root, modulus, sign, maximum or minimum value or remainder of a number. You can generate random numbers and make the sequence repeatable or arbitrary. You can specify angles in radians or degrees. You can calculate the sine, arcsine, cosine, arcosine, tangent, arctangent, cotangent, secant and cosecant. You can use the Boolean operators AND, OR, EXCLUSIVE OR and NOT. In other words, you can just plain "calculate" using the HP-75C in its "keyboard calculator" mode. Of course, all these calculating functions are also available in the "program" mode.

Using the Magnetic Strip-Cards

The built-in ability of the HP-75C to store information on or read from magnetic strip-cards puts it a big step ahead in the area of portable practicality. No matter how portable you make a computer, if you cannot load different programs into the unit quickly and conveniently (and without having to carry a separate briefcase full of paraphernalia to accomplish the task!), then you simply cannot take advantage of the computer's principle asset: that it is able to perform a wide variety of tasks on a moment's notice. With the HP-75C, all you need carry with you to load in a new program (or data), or to save programs/data, are a few long, slim, magnetic strip-cards!

These strip-cards are about 3/8'ths of an inch wide and 10 inches in length. Frankly, I would prefer to see these cards about three inches shorter. At 10 inches they tend to be a little top-heavy when carried in a shirt pocket. (Please excuse me for being so picky.) But, compared to the alternatives (you know – the briefcase full of gear), they are an absolute dream come true. Each card can hold up to 1.3 kilobytes of data or program instructions. In fact, there are two magnetic tracks on a card. Each track holds 650 bytes plus "header" information. It takes two passes through the card reader, once for each track, to transfer all the information that can be stored on a strip-card.

Of course, with that 48 kilobyte operating system in ROM, all of the strip-card operations are prompted, supervised and verified by the HP-75C. Suppose you want to load a new program into memory? OK, you just give the system command COPY CARD TO "FILENAME". This command creates a new file in memory, ready to receive the information that is on the card(s) you are going to process. At this point, the operating system proceeds to prompt you through each step of the card reading procedure. You are told to insert and align a card in the reader. (The reader is simply a slot provided on the front of the HP-75.) Next, you are directed to pull the card through the slot. A nice steady pull is all it takes. (If you don't do it right, the computer tells you whether you were too fast or slow!) After one track has been read, the computer "knows" how many other tracks need to be processed. It then prompts you for further inputs as necessary. The order in which a multi-track file is processed is irrelevant. The operating system takes care of organizing the various tracks into a complete file. With a little practice handling the cards, a 2.6K program (spread over a set of two strip-cards) can easily be loaded into memory in less than 30 seconds.

Writing information on cards is similarly prompted and guided by the operating system. Indeed, the process includes verifying the validity of the information stored on the card(s).

Oh, by the way, the HP-75C has several levels of program security. For instance, invoking the *private card* option prevents a user from listing, altering or duplicating a program. Invoking the *password* option can restrict a file to only those who know a user-specified password. And, a *protect* option allows you to safeguard strip-cards from being accidently overwritten. You can combine security options to increase the level of security, if desired. Such security provisions can, of course, provide a measure of comfort to those who need to create application packages that will be utilized by personnel to whom one does not wish to divulge proprietory algorithms, techniques or data.

Connecting to Peripherals

I have been suspicious for some time about the significance of Hewlett-Packard providing some rather powerful peripheral devices for their programmable HP-41 calculator. It has, frankly, seemed to me to be a bit of "overkill" to provide accessories such as the HP 82161A Digital Cassette Drive, capable of storing some 131,072 bytes of data, merely to provide storage for a calculator. After all, the drive costs roughly twice that of the HP-41 itself! Ah, but what if the real plan was to have a drive that could serve the calculator *as well as other devices.* And that, of course, was the plan.

You see, the HP-75C Portable Computer is equipped with HP-IL (Hewlett-Packard's Interface Loop). This makes it directly compatible with all those peripherals, such as the digital cassette drive just mentioned, and the HP-82162A Thermal Printer, that were recently made available for the HP-41.

The HP-IL system is a simple method (analagous, in some respects, to the old "teletype loop") for connecting a series of devices together. One device (the HP-75C, in this case) acts as the "talker," the others act as "listeners/relayers."

Part of the HP-75's operating system (remember that big 48K of ROM that I keep babbling about) contains all that is needed to quickly configure a system with peripherals. You just jumper the peripherals you want to use with HP-IL cables. (Two cables come with the HP-75, you get one more with each peripheral you purchase.) Then, you use an ASSIGNIO statement to assign device codes according to their serial position within the loop. From then on you can communicate with your peripherals through program or keyboard commands.

One other point I will make before moving on is this: TheHP-IL system is physically neat. You can, for instance, set up a "home base" system of peripherals, such as a video monitor, tape drive and printer. When at your home base, you simply plug in the two ends of the HP-IL connectors into the top of the '75 (this takes about five seconds). Presto. You are operating as a "desktop" system. Need to go on the road with your PC? Unplug those two connectors (takes about two seconds!) and you are on your way.

The Keyboard is Infinite

Well, very close to it. Almost every key is redefinable. And not merely to be another character, but to be your choice of another character, an entire expression or a command (or series of directives)! Not only that; any keyboard configuration you create can be stored as a file in memory. You can set up keys to call whole "keyboard configuration" files. Thus, you can instantly change the definitions of virtually all of the keys. The variations are endless. Believe me, you can create some highly sophisticated (to say nothing of "personalized") application programs with this type of key-defining capability. (It is analagous to

having a couple of hundred "softkeys" on a Sharp PC-1500, with the added ability of being able to instantly re-define all those keys!)

Powerful HP BASIC

I am only going to point out some of the special features about Hewlett-Packard's version of BASIC, after noting that, with 48K of ROM, they have provided a rather complete implementation of the popular language.

People nurtured on Microsoft BASIC will find it takes some getting used to dealing with strings in HP BASIC. For instance, you cannot define two-dimensional string arrays on the HP-75. You may, however, define a string to be any length (up to available memory). You can then create a function that subdivides a string so that each section can be treated as a separate element. The net result is essentially the same as manipulating a string array or matrix, but it takes some extra thought and care, at least until one gets used to the concept.

On the other hand, there are some string handling functions available in HP BASIC that simply are not available on many other machines. For instance, the UPRC$ function can convert lowercase strings to their uppercase representations. And, the POS string function can be used to locate the position of one string inside another.

A particularly strong area of HP BASIC is that it provides the option for users to create their own single-line or multiple-line functions. Thus, for example, if you frequently need to perform a particular operation, such as converting degrees Centigrade to Fahrenheit, you can just define the function. You give it a function name, such as FNC, and embody it in the form FNC(C), where the (C) indicates the parameter that is to be passed (Centigrade degrees, in this case) to the function. Any time you need to perform the conversion, you just invoke the FNC(C) function. It is a little neater than having to call a subroutine to perform such a procedure. What is more, variables used within a function definition are *local to that function.* However, variables used outside a function are available (global) to all functions. You can use function definitions to perform some pretty fancy and powerful programming.

The '75 has a rather interesting way of storing and recalling data in "external" files. (A file separate from the program being executed.) Access to such a file is essentially treated as though you were creating or reading from DATA statements! You can specify line numbers for the DATA statements as you create and write to them. Alternately, you can use default methods to have line numbers automatically assigned as needed. To recover information from such external files, you use READ # statements (as opposed to just plain READ statements when the data is within the same file as the program itself).

This method of processing external files is intriguing. Files created in this manner are accessible by the normal editing capabilities of the system. Thus, they are easily reviewed. In fact, they can be manually modified, if desired. I think most people will find this method of handling data in files easier than traditional computer methods. This is primarily because it provides for easier conceptualization of the external filing process.

The HP-75 also provides the ability to process this data serially or using random access techniques. Furthermore, special forms of the data-handling statements enable one to perform operations such as overwriting a particular line of data. It is thus possible to organize a random access file where the data elements vary in length! Other directives permit entire numeric arrays to be processed as a unit. Finally, ability to access plain text files under program control is provided, using similar types of DATA statements.

However, perhaps the most exciting capability of HP BASIC as implemented on the '75 is the ability to perform what HP terms a *program call*. Now, this "program call" capability is not the same as the old "chaining" method of loading in another program, with which you may be familiar. It is much more powerful!

First of all, the program CALL statement passes control to another program just as though that program were a subroutine. The calling program expects to eventually have control returned to itself.

However, the called program is executed as though it was an entirely separate unit. Variable values are not passed to the called program *unless* an external file is established for this purpose.

Never-the-less, the calling program *retains all its variable value assignments* (even those with names identical to ones used in the called program) and resumes operations with its retained variable values when control is returned!

The net result is this: You can create an entire series of programs, each to perform a specific operation or function. You build each of these programs in its own file and without regard to line numbering, variable assignments, etc., (unless you intend to pass specific parameters between programs). Then you can construct a master program (or programs) to call upon the various smaller programs as required to perform desired tasks.

And, if that form of operation in itself is not enough, *these program calls can be recursive!*

Now that, readers, is an example of providing real personal computing power.

There Is Still More

Of course, there are other (more mundane?) capabilities that one *might* expect in a PC. You can format your output to just about any layout desired using IMAGE and USING statements. These directives permit the mixing of numerical and string values. You can even specify the type of radix or separator symbol desired (European or American).

Even the beeper is programmable – and directly by specification of the frequency and duration. (All you music lovers: take note!)

Finally, I will mention that good debugging features are also provided. You can trace variables and/or program flow (branches). You can execute a program a step at a time. And, you can set up error traps.

Is There Anything Missing?

The only thing I could think of after spending a considerable amount of time reviewing this machine is that there is no provision (that I could find) for operating the unit's liquid-crystal display (LCD) in a bit-mapped mode. Ah well, most applications I have seen for utilizing bit-mapped graphics on a single-line display have been rather trivial. So, I am really not going to fault HP for leaving such capability out. You will just have to be satisfied with the 128 characters and their underlined counterparts that can be shown on the 32-character screen.

Styling

My compliments to HP's designers on the overall layout and design of this machine. It feels nice and it looks nice. The position of, for instance, the strip-card reader is practical and convenient. And, the accessability of the ROM drawers (right on the left-hand front of the unit) is splendid. One gets the distinct impression that the design group put a great deal of thought into this model. It will be appreciated by discerning users, I am sure.

By the way, whether designed for use in this position or not, I have found that the HP-75 fits quite comfortably (while touch-typing, at that) in my lap. If you travel a lot, such as by plane, the ability to work in this position (comfortably) could be of value.

The Manuals

You receive two manuals with the HP-75C. One is a huge, 360+ page, 8-1/2 by 11 inch volume entitled the *HP-75 Owner's Manual.* It is intended to serve as both a tutorial introduction and a reference manual. It is nicely organized, contains numerous illustrative examples, has a number of useful appendices, and a comprehensive index.

The second is named the *HP-75 Reference Manual.* It contains approximately 70 pages (8-1/2 by 11 inches). The purpose of this manual is to serve as a quick refresher and guide once you are basically familiar with the operation of the unit.

I rate this set of manuals: *superb.*

The Bottom Line

This gem sells for $995.00. I predict many discerning professionals will consider it a worthwhile investment. It is a pleasure to work with a tool that exhibits this level of overall quality and excellence of design.

Well done, HP. – *Nat Wadsworth, Editor*

ROBER MNEMONICS UPDATE

Norlin Rober has some additions and corrections to the material that appeared in the recent Special Edition of *PCN*. Plus, he has gleaned a lot of new knowledge. Here is a compilation of the latest findings by the Master Sleuth (on the Sharp PC-1500/Radio Shack PC-2):

1. Since the CPU does not use true indexed addressing, registers X, Y and U should more properly be referred to as Pointer Registers, not Index Registers.

2. The description of the operation of flag V contains an error. With reference to subtractions it should read ". . . is the *same* as that of the operand subtracted" rather than "is opposite to that of the operand subtracted."

3. The instruction designated INXY (opcode F5) does more than just increment X and Y. Prior to the incrementing, it transfers the contents of the address pointed to by X into the address pointed to by Y. The mnemonic should be changed to: STI (X) (Y).

4. The ADD instructions with opcodes EF, 4F, 5F and 6F are "Add *without carry*." That is, the condition of the carry flag *prior* to the addition has no effect. (The result of the addition, however, *does* affect the flags as usual.)

5. The condensed ROM map on page 4 of the Special Edition of *PCN* lists D5BF - DCAD as containing code. It should be D6BF - DCAD. Also, there is another brief lookup table located at E4E3 - E4EA.

6. Some additional instructions have been determined:

PWR DOWN (Code FD 4E) switches off the computer.
DSP OFF (Code FD C0) turns off the display.
DSP ON (Code FD C1) turns on the display.

7. New information on interrupts: A maskable interrupt (usable only when flag I is set, enabling the interrupt) pushes E and P onto the stack, then transfers control to the interrupt service routine that begins at E171 (the address stored in FFF8 & FFF9). The RTI instruction at the end of the interrupt routine pops P and E from the stack.

It appears that the BREAK key does not use the interrupt routine. When this key is pressed, bit 1 of alternate memory address F00B is set to 1. This address is checked routinely by instructions in ROM.

The manual paper advance apparently produces an interrupt.

It appears that the instruction represented by opcode FD CE sets the timer to produce a "timer interrupt" after a specified length of time. This time is related to the contents of the accumulator at the time FD CE is used, but the exact connection is not clear. If the accumulator contains zero, there is no interrupt at all. In other cases, the timer produces an interrupt after a length of time that depends on the accumulator contents. This length of time seems to be about 25 milliseconds at the longest. The timer interrupt routine begins at E22C, the address obtained from ROM locations FFFA & FFFB. It will occur only if flag I is set.

The FD CE instruction would presumably be followed by a WAI instruction to produce a programmable delay lasting until the interrupt occurred.

The FD DE opcode, not used in ROM, is similar to FD CE in effect, but the resulting delay times are different.

8. New information on inputting to the CPU from the keyboard: Connections to the 64 keys (exclusive of the ON key) form the electrical equivalent of an 8 by 8 matrix. The keyboard is polled as follows: The presence of a 1 bit in a particular position in the contents of alternate memory buffer address F00C specifies the corresponding column of the matrix. The CPU instruction LDA KB loads the accumulator with the byte having a zero bit in only the position corresponding to the row (of the specified column) in which a key is pressed. The byte loaded into A is then used to determine a location in the lookup table (FE80 - FEFF), from which the appropriate ASCII code is obtained.

The ROM routine that performs this operation begins at address E42C.

9. The power-up routine begins at E000, the address stored in bytes FFFE & FFFF of ROM.

10. Although the CPU has a non-maskable interrupt capability, it apparently is not used. The circuit diagram in the Service Manual shows the NMI pin as being grounded. The interrupt routine for NMI is located at E22B, which is the address stored in FFF8 & FFF9. This routine contains the RTI instruction.

11. According to the service manual, a third kind of interrupt, designated as a "Timer Interrupt" exists. It seems likely that the address stored in FFFA & FFFB, which is E22C, is where the Timer Interrupt begins. Note that at E22C there is a short routine ending with RTI and that it contains the code FD CE.

12. The addresses following the tokens in the list of BASIC words (ROM addresses C054 to C34D) are the starting addresses of the routines that execute the associated BASIC statements. For example the BEEP routine begins at E5C1. Note that words which may *not* begin a BASIC statement, such as THEN, TO, AND and OFF, are followed by CD89. This is the address at which the routine for ERROR 1 begins.

The routines executing BASIC statements end with CALL E2, where the stack pointer is reset and the next BASIC statement is read. If an error condition is to occur, CALL E4 (for ERROR 1) or CALL E0 (other ERROR) ends the BASIC statement routine.

The tokenized words representing functions, however, call on subroutines ending with RTS.

13. Revised versions of the ROM in the PC-1500 may differ from the ROM described above, particularly in regards to exact addresses of the various addresses.

14. Thanks to *James Stutsman* for pointing out that two separate ROMs may occupy the address space 8000 to BFFF at one time. The opcode B8, which I am giving the mnemonic ROM1, selects the ROM used by the printer/cassette interface. Opcode A8 with the mnemonic ROM2 selects the other. The "other" could be the RS-232 interface promised by Sharp.

15. You can upgrade the dissassembler program to include the items discussed in this column with the following lines:

```
345   LPRINT "STI (X) (Y)";:RETURN
476   LPRINT "PWR DOWN";:RETURN
592   LPRINT "DSP OFF";:RETURN
593   LPRINT "DSP ON";:RETURN
```

16. If you would like the printout of disassembled ROM to include the addresses of subroutines called from the base page, make the following modifications to the disassembler program:

A. End line 50 with GOTO 53 (instead of GOTO 54).

B. Add line 53 to the program as shown here:

```
53    LPRINT ",";:TAB 14:C=D+65280:D=PEEK C:GOSUB 24:
      D=PEEK(C+1):GOSUB 24:D=PEEK A
```

[Norlin indicates that he is getting quite a bit of mail. Please remember when writing to him or any author, that it is always a nice courtesy to include a S.A.S.E. (self-addressed, stamped envelope) if you would like a reply. – N.W.]

PC-1500/PC-2 RENUMBERING PROGRAM

Milt Sherwin, 8602D E. Amherst Drive, Denver, CO 80231, provides this sought after utility. Space limitations in this issue restrict us to the presentation of brief operating instructions and the program listing. However, Milt has prepared a nice article explaining how the program works. If you are interested, let *PCN* know and we will try to present the details in an upcoming issue.

You need at least a 4K memory module in your PC to use this program. Use the program as follows:

1. Clear memory then load the program.

2. If you wish to renumber a previously developed program, then MERGE it into memory at this point, *before* trying to run the renumbering program. If you are going to develop (key in) a program that you may decide to renumber later, then immediately execute the renumbering program by going to the RUN mode and issuing a RUN command. This operation simulates a MERGE and protects the renumbering program from itself!

3. Develop your program. Assign a label to the start of the program under construction or modification so that you can test it by reference

to the label. This is necessary because the renumbering program simulates the MERGE operation. You will not be able to access your new program if you do not give it a label.

4. Anytime you want to renumber the program under development, go to the RUN mode and issue a RUN command. This activates the renumbering program. Respond to the prompt: ST,IC,EN,LN with the appropriate values as indicated here:

ST = Starting line number (defaults to first line of user's program).
IC = Line increment (defaults to an increment value of 10).
EN = Ending line number (defaults to last line of user's program).
LN = Initial line number assignment for renumbering (default is 100).

Use a comma after inputting each value or if you want to use the default value. (Thus, entering three commas [, , ,] would result in the renumber program using all its default values.)

5. At the end of each RUNning of the renumbering program, it will ask if it should be "unlinked." Answer Y for yes if you no longer wish to use the renumbering program. Answer N for no if you plan further development work. Repeat step 4 as necessary.

6. Once the renumbering program has been unlinked, you may LIST or CSAVE the user's program in a normal manner. The renumbering program is no longer accessible.

7. When you are all through, use NEW0 to normalize your PC. (The renumbering program accesses various pointers at the machine language level. NEW0 assures that all pointers are re-initialized.)

```
100: IF PEEK 30821<
     >PEEK 30825AND
     PEEK 30822<>
     PEEK 30826THEN
     150
110: IF PEEK 30824+
     1=256THEN POKE
     30826,PEEK 308
     24-255:POKE 30
     825,PEEK 30823
     +1:GOTO 130
120: POKE 30826,
     PEEK 30824+1:
     POKE 30825,
     PEEK 30823
130: POKE 30823,
     PEEK 30825:
     POKE 30824,
     PEEK 30826
140: POKE PEEK 3082
     3*256+PEEK 308
     24,255:GOTO 52
     0
150: CLEAR :DIM A$(
     0)*26:C=1:P=&7
     050:R=&7150:M=
     PEEK 30825*256
     +PEEK 30826
160: INPUT "ST,IC,E
     N,LN ";A$(0)
170: FOR J=1TO LEN
     A$(0)
180: C$=MID$ (A$(0)
     ,J,1)
190: IF C$=","THEN
     LET C=C+1:GOTO
     250
200: ON CGOTO 210,2
     20,230,240
210: A$=A$+C$:GOTO
     250
220: B$=B$+C$:GOTO
     250
230: D$=D$+C$:GOTO
     250
240: E$=E$+C$
250: NEXT J
260: S=VAL A$:F=0:N
     =100
270: I=VAL B$:IF I=
     0THEN LET I=10
280: E=VAL D$:IF E=
     0THEN LET E=65
     279
290: L=VAL E$:IF L<
     >0THEN LET F=
     INT (L/256):N=
     L-F*256
300: PAUSE "WORKING
     . . ."
310: IF S=0THEN 370
320: GOSUB 530:IF B
     =STHEN 350
330: IF PEEK (M+C+3
     )=255THEN 520
340: M=M+C+3:GOTO 3
     20
350: IF L=0THEN LET
     F=A:N=D
360: GOTO 380
370: GOSUB 530
380: POKE M,F:POKE
     (M+1),N
390: POKE P,A:POKE
     (P+1),D:P=P+2:
     POKE R,F:POKE
     (R+1),N:R=R+2
400: IF PEEK (M+C+3
     )=255THEN 460
410: IF B=ETHEN 450
420: N=N+I:IF N<256
     THEN 440
430: N=N-256:F=F+1
440: M=M+C+3:GOTO 3
     70
450: M=M+C+3:GOSUB
     530:IF PEEK (M
     +C+3)<>255THEN
     450
460: P=P-&7050:N=M:
     M=PEEK 30825*2
     56+PEEK 30826
470: C=PEEK (M+2):
     GOSUB 570
480: M=M+C+3:Z=Z-1:
     IF Z<>0THEN 47
     0
490: INPUT "UNLINK
     RENUM? ";A$
500: IF LEFT$ (A$,1
     )<>"Y"THEN 520
510: POKE 30821,
     PEEK 30825:
     POKE 30822,
     PEEK 30826
520: END
530: C=PEEK M:D=INT
     (C/16):B=D*409
     6+(C-D*16)*256
     :A=C
540: C=PEEK (M+1):D
     =INT (C/16):B=
     B+D*16+(C-D*16
     ):D=C
550: C=PEEK (M+2):Z
     =Z+1
560: RETURN
570: A$="":J=M+3:K=
     0
580: A=PEEK J:G=
     PEEK (J+1)
590: IF A=&0DTHEN 7
     80
600: IF A=&F1AND G=
     &92OR G=&94OR
     G=&AEOR G=&AB
     THEN LET J=J+1
     :GOTO 620
610: J=J+1:GOTO 580
620: J=J+1:A=PEEK J
630: IF A<&30OR A>&
     39THEN 660
640: IF K=0THEN LET
     K=J
650: A$=A$+CHR$ A:
     GOTO 620
660: L=LEN A$:IF L=
     0THEN 580
670: D=VAL A$
680: FOR X=0TO P
     STEP 2
690: E=PEEK (&7050+
     X)*256+PEEK (&
     7050+X+1):IF E
     =DTHEN 720
700: NEXT X
710: GOTO 760
720: D=PEEK (&7150+
     X)*256+PEEK (&
     7150+X+1)
730: D$=STR$ D:R=
     LEN D$:IF R-L>
     0GOSUB 790:
     GOTO 760
740: IF R-L<0GOSUB
     830:GOTO 760
750: GOSUB 880
760: A$="":K=0:IF A
     =&2CTHEN 620
770: GOTO 580
780: RETURN
790: I=PEEK (30823)
     *256+PEEK 3082
     4:H=I
800: POKE (I+R-L),
     PEEK I
810: I=I-1:IF I>K
     THEN 800
820: GOTO 860
830: I=K:H=PEEK (30
     823)*256+PEEK
     30824
840: POKE I,PEEK (I
     -R+L)
850: I=I+1:IF I<H
     THEN 840
860: C=C+R-L:POKE (
     M+2),C:J=J+R-L
870: H=H+R-L:POKE 3
     0823,INT (H/25
     6):POKE 30824,
     H-INT (H/256)*
     256
880: FOR X=1TO R
890: A$=MID$ (D$,X,
     1):POKE K,ASC
     A$:K=K+1
900: NEXT X
910: RETURN

STATUS 1
                1935
```

RPN CALCULATOR FOR PC-1/PC-1211

Here is *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158,* with another one of his creative endeavors. This program converts the Radio Shack PC-1 or Sharp PC-1211 to a double-precision, Reverse-Polish Notation (RPN) calculator (similar in operation to Hewlett-Packard machines). The program permits addition, subtraction, multiplication and division. All inputs and outputs use scientific notation. There are four "guard digits" calculated with each operation, and the 20 most significant digits are displayed.

The "stack" consists of four registers, referred to as X, Y, Z and T. Register X uses variables A, B, C and D. Register Y uses E, F, G and H. Register Z uses variables I, J, K and L. Register T is stored in M, N, O and P. The first variable associated with each register contains the first 8 digits of the mantissa, the second variable holds the next 8 digits. The third variable holds the last 8 digits. The fourth variable stores the decimal exponent.

For those unfamiliar with RPN, its operation is described briefly as follows: Whan an input occurs, it is stored in register X, with the previous contents of X shifted into Y, the previous contents of Y into Z and those of Z into T. This shifting of the register contents is called a stack *lift*.

An arithmetic operation is performed using the contents of X and Y as operands. The calculated result is placed into register X. Such an operation is accompanied by a stack *drop*. That is, register Z is shifted into Y and T drops into Z (as well as remaining in T).

The four registers may be viewed as "stacked" on top of each other in the following form:

REGISTER T
REGISTER Z
REGISTER Y
REGISTER X

The RPN system is efficient as it is very convenient to use previously calculated results in further operations. The primary principle to keep in mind is that an operation (add, subtract, multiply or divide) is given *after* the numerical entries for that operation have been made and not *between* operands as is done with an algebraic calculator.

The following operations are performed using the PC when it is set to the DEF mode:

SHIFT/L Lift stack, with input to replace X. Note: If ENTER is pressed without any input being keyed, the previous contents of X remain there. This is useful, for example, in squaring a number as the number need not be entered twice.

SHIFT/N Non-lifting input. The input simply *replaces* the contents of X. No stack lift occurs.

SHIFT/X Exchange the contents of X and Y. Useful when it is desired to exchange the operands prior to division or subtraction.

SHIFT/A Add X and Y. Result in X, with stack drop.

SHIFT/S Subtract X from Y. Result in X, with stack drop.

SHIFT/M Multiply X by Y. Result in X, with stack drop.

SHIFT/D Divide Y by X. Result in X, with stack drop.

Inputs must be in scientific notation, including a decimal point after the first digit of the mantissa. Negative numbers should only have the minus sign entered in the first 8-digit block.

The prompt READY is displayed after numerical or operand inputs. Keying ENTER (alone) shows the current contents of register X as a 20-digit mantissa. Keying ENTER a second time displays the exponent.

Example: To calculate (2.176513 + 408.99716238)/(79.312294758 - .4851339176288), proceed as shown below.

With the program loaded satisfactorily, set the PC to the DEF mode.

(1) SHIFT/L. Enter inputs as follows:

1ST 8 DIGITS?	2.1765130
NEXT 8?	0
NEXT 8?	0
EXP?	0 (Note: Inputs in scientific notation.)

(2) SHIFT/L (again) and enter inputs of the second operand:

1ST 8 DIGITS?	4.0899716
NEXT 8?	23800000

Program *RPN Calculator for PC-1/PC-1211*

```
1  "L" M=I:N=J:O=K:P=L:I=E:J=F:K=G:L=H:E=A:F=B:
   G=C:H=D
2  "N" Q=↑E-4:R=↑E8:S=↑E-8
3  INPUT "1ST 8 DIGITS? ";A:A=↑E7A:B=0:C=0:D=0:
   INPUT "NEXT 8? ";B,"NEXT 8? ";C,"EXP? ";D
4  PRINT "READY":X=↑E-6B:Y=X-INT X+↑E-10*INT
   (CQ+.5:X=↑E-9*(ABS ↑E2A+INT X+INT Y:Y=↑E10*
   (Y-INT Y:Z=D
5  IF X=10 LET X=1:Z=Z+1
6  PRINT USING "##.#########";X*SGN A;USING
   "##########";Y:PRINT USING;"EXP ";Z:GOTO 6
7  "" E=I:F=J:G=K:H=L:I=M:J=N:K=O:L=P:GOTO 4
8  "X" W=A:X=B:Y=C:Z=D:A=E:B=F:C=G:D=H:E=W:
   F=X:G=Y:H=Z:GOTO 4
20 "S" A=-A
21 "A" U=SGN AE:IF U=0 LET A=A+E:B=B+F:C=C+G:
   D=D+H:GOTO "
22 Z=D-H:IF Z*↑E16+(ABS A-ABS E)R+B-F+(C-G)S LET
   W=ABS E:X=F:Y=G:GOTO 24
23 W=ABS A:X=B:Y=C:A=E:B=F:C=G:D=H:Z=-Z
24 IF Z > 25 GOTO "
25 IF Z > 7 LET Z=Z-8:Y=X+YS:X=W:W=0:GOTO 25
26 IF Z LET Z=10^-Z:Y=YZ+(XZ-INT XZ)R:X=INT XZ+
   (WZ-INT WZ)R:W=INT WZ
27 V=SGN A:A=ABS A+WU:B=B+XU:C=C+YU:GOTO 50
30 "M" V=SGN AE:A=ABS ↑E-3A:B=↑E-3B:C=10C:D=D+H:
   E=ABS EQ:F=FQ:W=A-INT A:X=B-INT B:Y=E-INT E:
   Z=F-INT F
31 T=W*INT E+Y*INT A:U=W*INT F+X*INT E+Y*INT
   B+Z*INT A:C=(BG+CF)QS+(U-INT U+XY+WZ)R+AGQ+
   BF+CEQ
32 B=INT A*INT F+INT B*INT E+INT U+(T-INT T+WY)R:
   A=INT A*INT E+INT T:GOTO 50
40 "D" V=SGN E/SGN A:D=H-D-1:H=ABS A+BS:A=ABS
   AQ:B=BQ:C=CQ:T=A-INT A:U=B-INT B:W=INT((ABS
   ER+F)/H)*Q
41 X=W-INT W:Y=U*INT W+X*INT B:E=(ABS E-INT A*
   INT W-T*INT W-X*INT A-TX)R+F-INT B*INT W-INT Y
42 F=G-(Y-INT Y)R-UXR-C*INT W-CX:X=INT((ER
   +F)/H)*Q:Y=X-INT X
43 C=(((E-INT A*INT X-T*INT X-Y*INT A-TY)R+F-
   INT B*INT X-U*INT X-Y*INT B-UY)R-CX)/H:A=↑E4W:
   B=↑E4X
50 C=CS+1:B=(B-1+INT C)S+1:A=A-1+INT B:B=(B-INT B)R
   :C=(C-INT C)R
51 IF A=0 IF B=0 IF C=0 LET D=0:GOTO "
52 IF A=0 LET A=B:B=INT C:C=(C-INT C)R:D=D-8:
   GOTO 52
53 IF A < ↑E7 LET U=1+INT LOG A:D=D+U-8:U=10^-U:
   C=CU:B=BU:A=AUR+INT B:B=(B-INT B)R+INT C:
   C=(C-INT C)R
54 IF A >= R LET A=.1A:B=.1B+(A-INT A)R:C=.1C+(B-
   INT B)R:B=INT B:A=INT A:D=D+1
55 A=AV:GOTO "
```

NEXT 8? 0
EXP? 2

(3) SHIFT/A, X and Y are added. The result is in X. You can view X by keying ENTER (after READY appears). It should be 4.1117367538000 000000 EXP 2.

(4) SHIFT/L. Now entering the denominator of the original expression:

1ST 8 DIGITS? 7.9312294
NEXT 8? 75800000
NEXT 8? 0
EXP? 1

(5) SHIFT/L (once more!) and enter:

1ST 8 DIGITS? 4.8513391
NEXT 8? 76288000
NEXT 8? 0
EXP? -1

Right at this point, the internal stack contents are:

REGISTER Z 411.17367538
REGISTER Y 79.312294758
REGISTER X .4851339176288

(6) SHIFT/S. This calculates Y-X and drops the stack one notch. You can examine the result of the subtraction by pressing ENTER after the READY prompt appears. The stack now contains:

REGISTER Y 411.17367538 (previously in Z)
REGISTER X 78.827160843712 (after subtracting)

(7) SHIFT/D. This performs the division operation to yield the final desired result for the original expression. Key ENTER to display it:

5.216142139 2893562452 EXP 0

When Keying in the Program

Remember, the character "↑E" stands for the EXPonent key, not the letter E.

Also, be sure and enter the program exactly as shown. A number of "tricks" to cram a lot of operations into the PC, such as implied multiplication, are used. Just enter the program *exactly* as shown, then try the example to see if you get the same results. If an error shows up, examine the line indicated in the error message for a keying mistake.

[So what else can you wring out of a little old PC-1? Nice work, Norlin. Some of you programming pro's might want to try a hand at converting this one for the PC-2/PC-1500. Looks like it would come in mighty handy in many applications. The speed of the PC-1500 should bring up results in the blink of an eye! – N.W.]

The POCKET COMPUTER NEWSLETTER is Available!

By Subscription Only: for a calendar year period (January – December). You get all issues published to date for the calendar year in which you subscribe, at the time you subscribe.

MC/VISA Phone Subscriptions: (203) 888-1946

☐ 1982 Regular Subscriber (Issues 11 – 20). $30.00 in U.S. (U.S. $36.00 to Canada. U.S. $45.00 elsewhere.)

☐ 1982/83 Subscriber (Issues 11 – 30). $60.00 in U.S. (U.S. $72.00 to Canada. U.S. $90.00 elsewhere.)

☐ 1983 Regular Subscriber (Issues 21 – 30). $36.00 in U.S. (U.S. $42.00 to Canada. U.S. $50.00 elsewhere.)

Orders must be accompanied by payment in full. We do not issue invoices for the POCKET COMPUTER NEWSLETTER. Thank you for your remittance.

Name: _____

Addr: _____

City: _____ State: ___ Zip: _____

MC/VISA #: _____ Expires: _____

Signature: _____

P.O. Box 232, Seymour, CT 06483

FROM THE WATCH POCKET

If you are interested in closely following the emergence of Hewlett-Packard's PCs, I suggest you send a large 9 by 12 inch, self-addressed envelope with 2 ounces of first class postage on it, to: **PPC,** *2545 West Camden Place, Santa Ana, CA 92704.* The organization puts out several journals of interest to HP-oriented PC enthusiasts. Sending in that stamped envelope addressed to yourself will reportedly bring you samples of the material put out by this organization.

Casio PC users might want to inquire about a newsletter being published by **Department 'G' Software,** *212 South Oak Street, Owatonna, MN 55060.*

Sharp Electronics will be providing U.S. deliveries of its new RS-232 interface beginning in November. The list price will be $225.00. The unit can be connected directly to the PC-1500 or the printer/cassette interface. It is reported to be capable of providing two-way, asynchronous, communication signals, programmable at rates of 50, 100, 110, 200, 300, 600, 1200 and 2400 baud. Other parameters, such as character size, parity and stop-bit length, are also programmable. A set of four AA nickel-cadmium batteries will power the unit for up to two hours in portable operation. The unit is also reported to contain firmware that converts the PC to a communications terminal. The system can then be connected (through an appropriate modem) into communications networks via telephone, connected to external devices such as full-sized printers or plotters, used to monitor instruments, talk with a desktop computer or drive a video display device.

Sharp is also due to begin deliveries of its ROM software modules this month. Mort Roserstein of **Atlantic Northeast Marketing,** *PO Box 921, Marblehead, MA 01945,* telephone (617) 639-0285, says he will have them as soon as they become available.

With the Casio FX-700P coming in at $99.95, how long do you suppose Radio Shack will try to hold a $149.95 price on the PC-1? I bet they move to give it *and* the PC-2 a substantial reduction in time for the Holiday Season this December.

By the way, I hear now that the price of the FA-3 cassette interface for the FX-700P will be $49.95. I am surprised. I figured a $99.95 PC would have, say, a $29.95 cassette interface. After all, even the "high priced" Radio Shack PC-1 can be equipped with a cassette interface at $29.95. Oh yes, word is that Casio *will* bring out a little printer for their newest PC. Probably won't be on the shelves, though, until the first quarter of '83.

Several people have pointed out that the PC-1500 has a CPU register designated as register W in the PC-1500 Service Manual. No machine codes for the use of this register were given in *PCN*'s Special Edition. That is right, they are missing at this point. But, as the saying goes, "People are working on it."

If anyone is still trying to load tapes for a PC-1 into a PC-2, *forget it!* Several people have complained that some advertisements seem to infer that "programs written for a PC-1 can be loaded directly into a PC-2." They have then spent hours trying to load tapes created for the PC-1. While technically the PC-2 is capable of executing the language statements of the PC-1, this can only be done if no programming shortcuts (such as implied multiplication) are used in the PC-1 version. But, irrespective of this, *you cannot load a PC-1 tape into a PC-2!* The tape recording characteristics are simply not compatible!

Martin Shapiro has been having a lot of fun building "humungous" (that is an acronym for "very large") programs for his PC-2. He buys TDK endless loop cassettes, (the kind that do not have any end of tape foil on them). Then he loads 'em up with programs. He can chain endlessly from program to program without pausing to fool around with the recorder controls. Do you see possibilities for your own applications from that idea?

While on cassettes, several readers have cautioned that attempting to use most microcassette recorders for storing programs requires use of top-of-the-line tape. The lower grade ones just can't seem to hack it.

Walter Tolbert, P.O. Box 1465, South Gate, CA 90280, telephone (213) 566-7449, wants to form a club of PC and HHC users in his area. Contact him if you are interested in forming such a group.

You can expect to see a flood of PC-2/PC-1500 software hitting the market in the next few months. – *Nat Wadsworth, Editor*